並列プログラムの動作評価に関する一検討

2R-2

安藤 津芳*、長谷川 晴朗*、長谷川 隆三**
*沖電気工業(株) **(財)新世代コンピュータ技術開発機構

1.はじめに

通信システムの世界では、並列ユーザ数・サービスの高度化/大規模化・高速のデータ通信の実現等を強く要望されており、並列性並びに高速性は克服すべき必須の課題として存在する。一方、一般情報処理分野においてもソフトウェアの高度化・複雑化・大規模化が進み並列処理計算機の研究が盛んであり、成果も上がってきている。故に、汎用の並列処理計算機上における、ソフトウェア開発用支援環境の確立は、今後のシステム構築のために重要な項目となる。そこで、この支援環境作成の一環として、特に並列プログラムのアルゴリズムの動作評価方式について検討を進めたので報告する。なお、今回我々が対象としたのは、負荷分散方式の疎結合マルチプロセッサシステム上に構築されたGHC実行環境である。

2.動作評価の位置付け

本来プログラム評価というのは、評価する側の立場や時期によって色々異なるものである。即ちあるものはその保守性や信頼性が優先され、別のものでは実行時間だけが優先され、さらには開発期間が優先される等の場合がある。このように種々な側面を持つプログラム評価ではあるが、並列処理化することの最大目標が一般的に速度の向上である以上、評価の指針として対象プログラムの実行効率を最も高めることを考える。

2.1 動作評価の目的

我々にとっての動作評価の目的というのは、先に示したような背景の下で、

「入力を色々変更し何度でも実行される対象プログラムが、対象計算機上で平均的に効率良く実行できるように改良するポイントを明確化すること。」

である。

即ちこれは、実行効率(処理時間・並列実行度・アルゴリズムの効率・通信待ち時間・中断時間・プロセス割付・優先度割付・共通データアクセス等)を、入力データの分布を基に評価し、より効率的なアルゴリズムに変更するための参考とすべき情報を得ることである。特に並列度と通信コスト間の対象プログラムでの実行時間に対するトレードオフ関係を明確にし、対象プログラムの入力データ特性を加味して最適な解の算出を行うものである。

2.2 動作評価手法

並列計算機の利点のみの計算機を理想計算機と位置付け(ex.プロセッサ数無限、通信遅延無し等)、この計算機における対象プログラムの要求時間T_mを、極小時間$\triangle t$で分割する。即ち$T_m = N \cdot \triangle t$と考える。そして$\triangle t_i$時点での並列度を$P_i$と定義すると、この対象プログラムの逐次計算機での要求時間Sは

$$S = \triangle t \cdot \sum_{i=1}^{N} P_i$$

で表される。また平均並列度P_mを次のように定義する。

$$P_m = S / T_m = (\sum_{i=1}^{N} P_i) / N$$

すると理想並列計算機と逐次計算機との関係はこれらを用いると次のように表せる。

$$S = P_m \cdot T_m$$

ここで示したそれぞれのデータは仮想環境で実行させることで収集可能な情報である。

次に、対象とされる実計算機環境を考えた場合、再現性の問題はあるにせよ統計的見地から各地点での状態推移確率を決定し、ある程度の確からしさで平均応答時間T'_mのデータは収集できる[3]。そして平均並列度P'_mを算出する。ここでのT'_mとP'_mの関係は、一般的には通信遅延や同期処理のためにP'_mがある大きさになる迄はP'_mが大きくなるに従ってT'_mは減少するが、それを越えると逆に増加を始めるというものである。そこでこの曲線を近似し最適解を求めるために、試行錯誤することになるが、その効率を良くするために静的解析にて次の情報を予め求める。

- 入力Inと各瞬間の並列度Pと通信コストCの実行時間Eとの関係$E = f(In, P, C)$の具体化。
- 非決定要素の抽出
- 入力データ分布とアルゴリズムからの最適試験データの算出

そして、設計者はこれらの関係やデータを参考にして、アルゴリズムの変更や各種試験データによる試行によりターゲットマシン及びターゲット処理に最適なプログラムを作成する。

A Study on an Estimation Method of Program on Parallel Computers
Tsuyoshi ANDO*, Haruo HASEGAWA*, Ryuzo HASEGAWA**
* Oki Electric Industry Co., Ltd. ** Institute for New Generation Computer Technology

3.動作評価実現方式

本動作評価を行うための実現方式として、我々は図1の様な構成の支援環境を考えた。

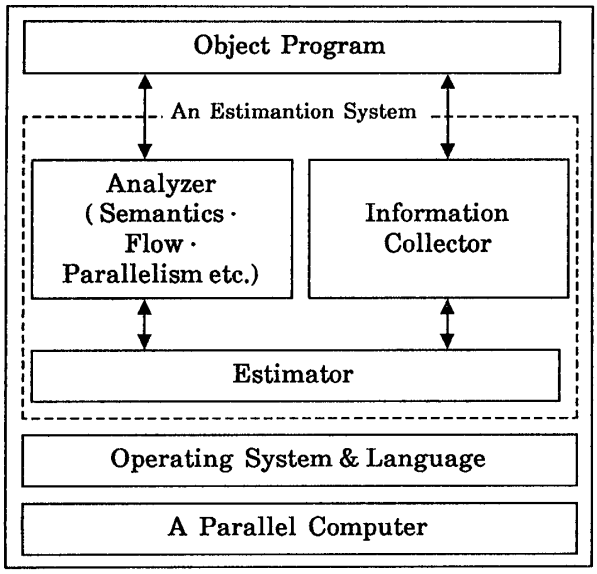

図1　動作評価支援環境構成図

3.1 Analyzer

本Analyzerについては、別論文[1]で報告しているので、ここでは簡単な説明にとどめる。Semanticsだけでなく Flow や Parallelism の分析を実現するための技術として、本Analyzerは非同期並行システムのモデル化に優れたペトリネットを用いている。即ち、仕様もしくはプログラム言語から、制御に関する情報を取り出しペトリネットへ変換し、ペトリネットを解析することでプログラム分析を行うものである。

3.2 Information Collector

Information Collectorとはその名の通り情報収集処理である。収集情報としてはこれまでに示した評価項目を満たすためのものであり、具体的には時系列なリダクションや通信の情報である。そしてこの情報を収集するための機構として我々はMeta-Interpreterの技術[2]を使うことにした。即ち、必要とする情報一種類を収集するMeta-Interpreterを複数作成し、評価対象とするプログラムに対して、知りたい情報を表示するのに必要なMeta-Interpreterを階層的に使用することで実現する方法である。

この方法の特徴としては、

・個々のInterpreterは単機能のため、実現が容易
・情報の取捨選択が簡単
・Interpreter方式のため実行を自由に制御できるので、ユーザの望む条件を実現可能

等が考えられるが、その一方で、

・Interpreterによる影響の明確化

という課題を対処しなければならない。そこでこれに関しては現在我々の選択した環境下で評価・検討中である。

またここで処理の非決定性要素を持つ再現性の乏しい並列プログラムにおいて、時系列な一過性の実行情報がどの程度有効かという疑問が生じるが、平均応答時間と真の要求時間の関係[3]から、近似解が算出できる。また、次のようなことも考えられる。

・一度動いたからと言って必ずそうなるわけではないが、一つの指針となりえる情報である。
・そのマシン条件での動作を統計的にある程度知ることが可能。
・総ての動作が非決定ではないので、設計者意図とイメージ照合が可能。
・情報量と有効性に関しても設計者の意図で自由に制御可能。

以上の理由により、設計者にとっては直接的・間接的に十分に有効な情報であると考える。

3.3 Estimator

Analyzer や Information Collector によって収集された情報というのは、あくまでもその解析結果又は実行結果に過ぎず、これだけであればその評価は総て設計者が行なわなくてはならず設計者にとって大変な作業であると共に評価が主観的になり、真に効率的なアルゴリズムの実現ができない可能性がある。そこで、Estimatorを、設計者に対する負荷を少しでも軽減すると共に、より客観的な評価ができるようにするために作成する。

具体的には、解析結果及び実行結果のデータベースとこれまでの変更内容と実行効率の変化の関係を示す知識ベースを基に、設計者に対して、アルゴリズムの変更点・内容をアドバイスするものと考えている。従って、初期の段階では、有効なアドバイスはできないが、設計者が試行錯誤するに従ってより有効に使えるものとなるであろう。また、種々のプログラムを対象にすることにより、より有効なツールに発展する可能性を持つ。

4.まとめ

今回我々は、並列プログラムの動作評価なかでも特にアルゴリズムの評価法について、一案を示した。現在この案に基づき我々の考えを立証すべく試作評価中である。

なお、本報告は第五世代コンピュータ研究の一環として行われたものである。

[参考文献]

[1] 前田他："ペトリネットによる並列処理プログラミング動作解析に関する一考察," 情報処理学会第40回全国大会

[2] 田中他；"変数管理をするGHCの自己記述," 信学技報 Vol. 88, No 52, pp 41-49

[3] Bin Qin, et al. : " A Model to Estimate average Response Time of Parallel Programs," Proc. COMPSAC89, pp. 67-74, Sep. 1989